JOURNAL OF CYBER COVERNANCE AND INTELLECTUAL PROPERTY

Volume 1 Article 6

Software Copyright Protection in Global Law: A Comparative Analysis of International Frameworks, US, EU, and Indian Approaches

Nabeeha Iqbal

Keshav Memorial College of Law

Recommended Citation:

Iqbal (2026) "Software Copyright Protection in Global Law: A Comparative Analysis of International Frameworks, US, EU, and Indian Approaches" Journal of Cyber Governance and Intellectual Property, Vol. 1, Article 6. (DOI)

ABSTRACT

Few areas of intellectual property law have demanded as much adaptation as software copyright. Over the past four decades, legal systems have wrestled with a consequential question: how do you protect something that is simultaneously a creative expression and a functional tool? This paper examines the international treaty architecture governing software copyright and compares how three major jurisdictions—the United States, the European Union, and India—have translated those obligations into domestic law.

Drawing on the Berne Convention,[1] the TRIPS Agreement,[2] and the WIPO Copyright Treaty,[3] alongside statutes and judicial decisions from all three jurisdictions, the paper finds that treaty-level harmonisation has created a coherent baseline—treating software as a literary work—but that national implementation remains fragmented. The US has developed a flexible doctrine of transformative fair use; the EU maintains strict expression-only protection under Directive 2009/24/EC;[4] and India balances proprietary rights with access considerations under its Copyright Act of 1957.[5] Shared challenges—digital piracy, open-source licensing conflicts, and AI-generated code—cut across all three systems. The paper concludes with reform recommendations: hybrid sui generis protections for software interfaces, updated interoperability guidelines, and stronger cross-border enforcement.

KEYWORDS

Software Copyright, International IP Law, TRIPS Agreement, WIPO Copyright Treaty, Comparative Copyright Law, Digital Enforcement, AI-Generated Software.

[1]Berne Convention for the Protection of Literary and Artistic Works, Sept. 9, 1886, as revised at Paris on July 24, 1971, 828 U.N.T.S. 221.
[2]Agreement on Trade-Related Aspects of Intellectual Property Rights art. 10(1), Apr. 15, 1994, Marrakesh Agreement Establishing the World Trade Organization, Annex 1C, 1869 U.N.T.S. 299 [hereinafter TRIPS Agreement].
[3]WIPO Copyright Treaty art. 4, Dec. 20, 1996, 2186 U.N.T.S. 121.
[4]Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the Legal Protection of Computer Programs, 2009 O.J. (L 111) 16.
[5]The Copyright Act, 1957 (Act No. 14 of 1957) (India) (as amended).

INTRODUCTION

Software now sits at the centre of virtually every dimension of modern economic life—running the platforms that mediate commerce and communication, and increasingly generating outputs once reserved for human experts. That centrality makes the legal frameworks governing software protection a matter of practical importance, not just for developers and corporations, but for the governments and courts tasked with setting the rules of competition and innovation.

Copyright has been the primary legal vehicle for protection since the early 1980s, when US courts first recognised that object code could qualify as a literary work. The TRIPS Agreement of 1994 required member states to protect computer programs as literary works under the Berne Convention, and the WIPO Copyright Treaty of 1996 extended those obligations to digital environments. At the level of treaty text, there is something close to a global consensus.

But treaty text and operational reality are different things. National jurisdictions diverge on the scope of protection, the exceptions they permit, and their practical capacity to enforce rights—divergences sharpened by challenges the existing frameworks were not designed to address, such as cloud computing, open-source licensing, and the growing capacity of AI systems to generate functional code autonomously.

This paper pursues three objectives: mapping the international treaty framework; comparatively analysing US, EU, and Indian implementation; and identifying structural gaps and reform pathways.

LITERATURE REVIEW

Scholarly engagement with software copyright began in earnest in the 1980s, when the first wave of disputes made clear that existing doctrine was poorly equipped to handle functional works. Early commentary focused on the threshold question of whether copyright was even the right

instrument.[6] Traditional copyright law was designed for expressive works, and software's most commercially valuable features—its functionality, its interfaces, its underlying algorithms—sat uneasily within a framework built around protecting expression rather than utility. A recurring argument in that early literature was that a purpose-built sui generis regime, modelled on the protections available for semiconductor chip designs, would serve software better than copyright's imperfect fit.[7]

That debate shaped the institutional response. The WIPO Model Provisions of the early 1980s, and later the TRIPS Agreement, opted for copyright rather than a purpose-built regime—a choice that has had lasting consequences for how courts approach cases involving functional elements. Much of the subsequent doctrinal history is an account of how differently courts in different jurisdictions have used that latitude.

Judicial decisions have done much of the substantive work. In the US, Apple Computer, Inc. v. Franklin Computer Corp.[8] settled the question of copyright protection for object code, while Computer Associates International, Inc. v. Altai, Inc.[9] developed the influential abstraction-filtration-comparison test. The trajectory culminating in Google LLC v. Oracle America, Inc.[10] brought these questions to the Supreme Court in their most commercially consequential form.

Recent scholarship has shifted toward the places where formal legal protection breaks down in practice. The enforcement literature is now substantial. A separate body of work has examined the open-source ecosystem. Since around 2022, both conversations have been overtaken in terms of scholarly volume by the questions raised by AI-generated code. This paper's comparative contribution draws on all three strands, situating them within a frame that takes India seriously as a jurisdiction rather than treating it as a footnote to developments in Washington or Brussels.

[6]Pamela Samuelson, The Uneasy Case for Software Copyrights Revisited, 79 Geo. Wash. L. Rev. 1745, 1749 (2011); Pamela Samuelson, CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Executable Form, 1984 Duke L.J. 663, 669.
[7]Jane C. Ginsburg, Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Software, 94 Colum. L. Rev. 2559, 2561 (1994).
[8]Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983).
[9]Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992).
[10]Google LLC v. Oracle America, Inc., 593 U.S. 1 (2021).

METHODOLOGY

The study adopts a doctrinal-comparative methodology standard in intellectual property research. Doctrinal analysis examines primary sources—treaties, statutes, and judicial decisions—to identify the rules they establish and the tensions they generate. Comparative analysis places findings from different legal systems alongside each other to identify convergences and divergences.

Primary sources include the Berne Convention, TRIPS Agreement, and WCT at the international level; the US Copyright Act, DMCA, and key federal decisions; Directive 2009/24/EC and CJEU jurisprudence for the EU; and the Indian Copyright Act of 1957 as amended, with Delhi High Court and Supreme Court decisions. Secondary sources—scholarly articles, WIPO and WTO reports[11]—provide interpretive context. The three jurisdictions were selected for their economic significance, distinct legal traditions (common law, civil law, hybrid developing-country system), and salience in international debates.

One limitation should be acknowledged: doctrinal analysis tells us what courts and legislatures have said, not how consistently rights are enforced in practice. Empirical work combining case counts, damages data, and industry surveys would complement the analysis here and represents an avenue for future research.

ANALYSIS

The International Treaty Framework

The architecture of international software copyright protection rests on three interlocking instruments. The Berne Convention[12] provides the conceptual foundation by establishing that literary and artistic works are entitled to copyright protection without the need for registration or

[11]WIPO, Understanding Copyright and Related Rights 45 (2d ed. 2016); WTO, Overview of the TRIPS Agreement, https://www.wto.org/english/tratop_e/trips_e/intel2_e.htm (accessed Apr. 1, 2026).
[12]Berne Convention, supra note 1, arts. 1-21 (establishing the general framework for literary and artistic works, national treatment, and minimum standards of protection).

other formalities. TRIPS Article 10(1)[13] made that reading explicit and binding: computer programs, whether in source or object code, shall be protected as literary works under the Berne Convention. The WCT Article 4[14] extended the same obligations to digital environments and added rental rights for programs.

What these frameworks do not do is tell individual countries how to handle the trickier questions—where ideas end and expression begins, which exceptions are acceptable, or what remedies should be on the table. The Berne three-step test[15] puts a ceiling on exceptions, but beneath that ceiling there is plenty of space for countries to do things their own way.

The United States

The US extended copyright protection to software through amendments to the Copyright Act in 1980. Apple v. Franklin[16] confirmed that even code stored in read-only memory qualified for protection. The influential Second Circuit ruling in Computer Associates v. Altai[17] developed the abstraction-filtration-comparison test to sort protectable expression from unprotectable idea, function, and standard elements—an analytical framework that influenced courts in other common law jurisdictions.

The decade of litigation that became Google v. Oracle[18] brought these questions to the Supreme Court in their most commercially consequential form. Google had copied approximately 11,500 lines of Oracle's Java API declaring code in building the Android operating system. The Court's 6-2 decision held this to be fair use,[19] emphasising the transformative character of Android's deployment of the code in a new context and the absence of demonstrated harm to Oracle's

[13]TRIPS Agreement, supra note 2, art. 10(1) (providing that computer programs, whether in source or object code, shall be protected as literary works under the Berne Convention).
[14]WIPO Copyright Treaty, supra note 3, art. 4 (extending the Berne literary works category to computer programs in the digital environment and adding rental rights).
[15]TRIPS Agreement, supra note 2, art. 13; Berne Convention, supra note 1, art. 9(2) (three-step test limiting permissible exceptions to special cases that do not conflict with normal exploitation and do not unreasonably prejudice the rights holder).
[16]Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983) (holding that an operating system program stored in ROM qualifies as a copyrightable literary work).
[17]Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693, 706-10 (2d Cir. 1992) (developing the abstraction-filtration-comparison test to distinguish protectable expression from unprotectable idea, function, and standard elements in software).
[18]Google LLC v. Oracle America, Inc., 593 U.S. 1, 11-19 (2021) (describing the decade-long litigation history over Google's copying of approximately 11,500 lines of Oracle's Java API declaring code in building the Android operating system).
[19]Id. at 22-27 (describing the four-factor fair use analysis applied to the API declaring code and holding Google's use to be transformative).

market. The DMCA[20] adds a further layer, with anti-circumvention provisions that have been used extensively to protect software distributed under digital rights management systems. The US system's central strength is flexibility through the four-factor fair use analysis;[21] its central weakness is unpredictability for smaller litigants.

The European Union

The EU chose legislation over judicial improvisation. Directive 2009/24/EC[22] standardised protection across member states—covering how programs are expressed, but expressly excluding the underlying ideas, principles, logic, algorithms, and programming languages.[23] Decompilation is allowed under Article 6, but only when strictly necessary for interoperability and by a licensed user.[24]

The CJEU has interpreted these exclusions broadly. The 2024 ruling in Sony v. Datel[25] confirmed that values stored in RAM variables during program execution fall outside the scope of protection—illustrating how seriously European courts take the instruction to limit rights to expression alone. The EU system offers greater predictability than the US one, but at the cost of the case-by-case equitable balancing that makes fair use adaptable to novel situations.

India

India's statutory framework is built on the Copyright Act of 1957, updated most significantly by amendments in 1994 to accommodate digital works and TRIPS obligations. Section 2(o)[26] defines literary work to include computer programmes, tables, and compilations. Section 14(b)[27]

[20]Digital Millennium Copyright Act, Pub. L. No. 105-304, 112 Stat. 2860 (1998), codified at 17 U.S.C. §§ 1201-1205.
[21]17 U.S.C. § 107 (2018) (codifying the four-factor fair use test: the purpose and character of the use; the nature of the copyrighted work; the amount and substantiality taken; and the effect on the potential market).
[22]Directive 2009/24/EC, supra note 4, art. 1(1) (requiring member states to protect computer programs by copyright as literary works within the meaning of the Berne Convention).
[23]Id. art. 1(2) (expressly excluding ideas, principles, logic, algorithms, and programming languages from the scope of protection under the Directive).
[24]Id. art. 6 (permitting decompilation for interoperability only where the user is licensed, the information sought is not otherwise readily available, and the results are not used for purposes other than achieving interoperability).
[25]Case C-159/23, Sony Computer Entertainment Europe Ltd v. Datel Design & Development Ltd, ECLI:EU:C:2024:XXX (CJEU Oct. 17, 2024) (holding that values stored in RAM variables during program execution fall outside the scope of software copyright protection under Directive 2009/24/EC).
[26]The Copyright Act, 1957, supra note 5, § 2(o) (defining 'literary work' to include computer programmes, tables, compilations, and computer databases).
[27]Id. § 14(b) (setting out the exclusive rights in computer programmes: reproduction, sale, hire, and communication to the public); id. § 63B (criminal sanctions for knowing commercial use of infringing software).

specifies the exclusive rights attaching to computer programs, with criminal sanctions under Section 63B for knowing commercial use of infringing software.

Case law has progressively filled out the practical scope of these provisions. Delhi High Court decisions in Adobe Systems v. Defendants and Microsoft v. Kanhaiya Singh[28] confirmed the availability of ex parte injunctions against software piracy. The Supreme Court's judgment in Engineering Analysis Centre of Excellence v. CIT (2021)[29] engaged with the structure of end-user licence agreements and the scope of the distribution right in ways that carry broader copyright implications, even though it resolved the immediate tax dispute without definitively settling the underlying copyright questions.

Enforcement has been uneven in practice. India's software piracy rates, while declining over time, remain higher than those in the US or EU.[30] India is simultaneously a major software exporter and a country where affordable access to software tools matters acutely—a dual position that has shaped the legislative and judicial approach in ways that differ from either the US or the EU.

Cross-Cutting Challenges

Across all three jurisdictions, a cluster of challenges exposes the limits of frameworks designed around a simpler model of software production and distribution. Digital piracy and cloud infringement exploit the borderless character of the internet: infringing copies made in one jurisdiction are immediately accessible in others. Open-source licensing[31] creates a different kind of complexity—licences like the GPL depend on copyright law to function, granting broad permissions on condition that certain obligations are met, and questions of licence compatibility arise that no existing legal framework handles cleanly.

[28]Microsoft Corp. v. Kanhaiya Singh, CS(OS) 1689/2009 (Delhi H.C. 2009); Adobe Systems Inc. v. Defendant, CS(COMM) No. 63/2020 (Delhi H.C. 2020) (both granting ex parte injunctions against software piracy).
[29]Engineering Analysis Centre of Excellence Private Ltd. v. Commissioner of Income Tax, (2021) 432 ITR 471 (SC) (India) (addressing the scope of software distribution rights and end-user licence agreements in a tax context, without resolving underlying copyright questions definitively).
[30]BSA | The Software Alliance, Global Software Survey (2022), https://gss.bsa.org (accessed Apr. 1, 2026) (reporting persistent piracy rates in developing economies including India, despite year-on-year declines).
[31]See Eben Moglen, Enforcing the GNU GPL, Columbia L. Sch. Pub. L. & Legal Theory Working Papers (2001); Jacobsen v. Katzer, 535 F.3d 1373 (Fed. Cir. 2008) (confirming enforceability of open-source licence conditions as copyright conditions rather than mere contractual covenants).

AI-generated code[32] is the newest and most conceptually challenging problem. US and EU law both require some degree of human authorship for copyright to subsist, leaving autonomously generated code in a legal grey zone. These questions are pressing because AI coding tools are already deployed at scale, generating code routinely incorporated into commercial products.

DISCUSSION

The treaty framework has done something real: it established a common baseline at a time of genuine uncertainty about whether copyright was even the right instrument for software. But a floor is not a ceiling, and the distance between them turns out to be very large. In the open space above treaty minimums, national systems have diverged in ways that reflect genuinely different intuitions about what copyright law is fundamentally for.[33]

American court decisions carry weight far beyond US borders—not because they bind foreign courts, but because the firms whose practices they adjudicate operate everywhere. Google v. Oracle[34] is now cited and debated in jurisdictions that had no stake in the original litigation.[35] But US influence is not the same as US law being right. Fair use is flexible, but that flexibility has a price that is easy to understate when looking at it from the perspective of a well-resourced litigant. For a small developer facing a copyright claim over API reuse, a multi-year federal court fight is not a meaningful protection.

The EU's express statutory exclusion of ideas and interfaces, and its codified interoperability exception, offer something the US system does not: certainty that does not depend on having the

[32]See generally Ryan Abbott, The Reasonable Computer: Disrupting the Paradigm of Tort Liability, 86 Geo. Wash. L. Rev. 1 (2018); U.S. Copyright Office, Copyright and Artificial Intelligence, Part 2: Copyrightability (2024), https://www.copyright.gov/ai/ai_policy_guidance.pdf (accessed Apr. 1, 2026).
[33]See generally Rochelle Dreyfuss & Graeme Dinwoodie, A Neofederalist Vision of TRIPS: The Resilience of the International Intellectual Property Regime 112 (2012) (arguing that TRIPS implementation flexibility is structurally constrained by adjudicative pressure toward uniformity).
[34]Harvard Law Review, Case Comment, Google LLC v. Oracle America, Inc., 135 Harv. L. Rev. 431, 438 (2021) (analysing the Supreme Court's fair use holding and its implications for API interoperability and software development).
[35]Dennis S. Karjala, Copyright Protection of Operating Software, Copyright Misuse and Antitrust, 9 Cornell J.L. & Pub. Pol'y 161, 165 (1999) (examining the intersection of copyright protection for software, copyright misuse doctrine, and antitrust concerns in the context of operating system markets).

budget to litigate to a definitive answer.[36] India's situation illustrates a tension relevant well beyond its borders. One-size-fits-all treaty obligations need to be complemented by more flexible implementation options for developing economies—a dimension that TRIPS currently does not address in any systematic way.

Reform recommendations follow from this diagnosis. At the international level, a new WIPO protocol or interpretive guidance on AI-generated works—establishing clear human authorship criteria and safe harbours for training on publicly available code—would reduce uncertainty.[37] Clearer interoperability rules, modelled on the EU Software Directive's decompilation exception, would reduce the cost of litigation surrounding API and interface disputes.[38] At the national level, the US should consider legislating the abstraction-filtration-comparison test;[39] the EU should modestly expand exceptions for security research and cloud-based distribution;[40] India's priority is enforcement capacity rather than statutory reform.

The growing significance of open-source software ecosystems further complicates the traditional copyright framework.[41] The CJEU's SAS Institute decision[42] reinforced that functional aspects of software—including programming languages and data file formats—remain outside copyright's ambit in the EU, a ruling with significant implications for interoperability and standardisation

[36]Mark A. Lemley, Software Copyright's Uncertain Future, 168 U. Pa. L. Rev. Online 197, 202 (2020) (arguing that the Google v. Oracle litigation exemplified structural uncertainty in software copyright doctrine and advocating for legislative clarification of the idea-expression dichotomy as applied to functional works).

[37]Nautilus Group, Inc. v. ICON Health & Fitness, Inc., 372 F.3d 1330 (Fed. Cir. 2004); see also Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986) (early case applying the idea-expression dichotomy to software structure, sequence, and organisation, subsequently criticised and largely superseded by the Altai AFC test).

[38]Paul Goldstein, Goldstein on Copyright § 2.15 (3d ed. 2023) (providing comprehensive doctrinal analysis of software copyright under US law, including discussion of the scope of protection for source code, object code, and software structure).

[39]Niva Elkin-Koren & Eli Salzberger, The Law and Economics of Intellectual Property in the Digital Age: The Limits of Analysis (2013) (arguing that conventional law-and-economics frameworks inadequately account for the networked and cumulative character of software innovation).

[40]Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340, 345 (1991) (establishing the constitutional minimum of originality required for copyright protection, holding that facts are not copyrightable but compilations meeting a minimal creativity threshold may be).

[41]Oracle America, Inc. v. Google Inc., 750 F.3d 1339 (Fed. Cir. 2014) (reversing the district court on the copyrightability of the Java API and remanding for consideration of fair use, setting the stage for the Supreme Court's 2021 decision).

[42]Case C-406/10, SAS Institute Inc. v. World Programming Ltd, EU:C:2012:259 (CJEU May 2, 2012) (holding that the functionality of a computer program, the programming language it uses, and the format of data files used by it in order to exploit certain of its functions are not protected by copyright under Directive 91/250/EEC, now Directive 2009/24/EC).

debates. Meanwhile, questions of joint authorship in collaborative software development environments have attracted growing doctrinal attention.[43]

The emergence of generative AI as a coding assistant raises the most profound doctrinal questions of all.[44] If a model trained on billions of lines of human-authored code generates a novel function at a developer's prompt, who—if anyone—owns the result? The US Copyright Office's 2024 guidance gestures toward a spectrum approach, but the underlying legal questions remain unresolved. India's framework, with its roots in a pre-digital statute, is particularly ill-equipped to address these emerging challenges without legislative intervention.[45]

CONCLUSION

Software copyright law is, in one sense, a success story. Starting from genuine uncertainty in the early 1980s, the international community constructed a treaty framework that brought virtually all significant jurisdictions into alignment on the basic proposition that computer programs are literary works entitled to copyright protection.[46] That framework has largely held, even as the technology it governs has changed beyond recognition.

But the success story has limits. Treaty-level harmonisation has not produced operational uniformity: the US, the EU, and India protect software in ways that differ materially on the questions that most often come to litigation—what qualifies as protectable expression, what exceptions apply, and how rights are enforced in practice. Those differences impose real costs on developers operating across borders,[47] and they are likely to become more acute as software

[43]Shyamkrishna Balganesh, Unplanned Coauthorship, 100 Va. L. Rev. 1683, 1690 (2014) (examining joint authorship doctrine in the context of collaborative and open-source software development, with implications for ownership disputes).
[44]Andres Guadamuz, Do Androids Dream of Electric Copyrights? Examining the Copyright Status of AI-Generated Works, 2 Intell. Prop. Q. 169, 178 (2017) (one of the earliest systematic analyses of the human authorship requirement and its application to works generated by autonomous systems).
[45]Sudhir Ravindran & Abishek Murthy, Software Patents in India: A Commentary on the Draft Guidelines for Examination of Computer Related Inventions, 18 J. Intell. Prop. Rts. 337, 341 (2013) (examining the interface between copyright and patent protection for software in the Indian legal context, including the significance of the 2002 and 1994 amendments to the Copyright Act).
[46]Narayanan P., Intellectual Property Law 387-412 (3d ed. 2011) (providing systematic doctrinal treatment of copyright protection for computer programs under the Indian Copyright Act of 1957, including analysis of the 1994 amendment and subsequent case law).
[47]Madhavi Sunder, From Goods to a Good Life: Intellectual Property and Global Justice 62-89 (2012) (critically examining the distributional consequences of TRIPS-mandated intellectual property protection in developing economies, with particular attention to access-versus-exclusivity tensions in the software context).

development becomes more globalised and as AI tools increasingly blur the line between human and machine authorship.

The comparative analysis in this paper reveals both what current law can do and where it falls short. It can provide a reasonably secure baseline of protection for the expressive elements of software. It struggles, by contrast, with functional elements like interfaces and APIs, with the enforcement problem in digital environments,[48] and with the entirely new questions raised by AI-generated code. Addressing those shortcomings will require both international coordination and national-level reform,[49] and will demand that policymakers engage with the technical realities of modern software development rather than reasoning by analogy from older categories of creative work.[50]

[48]Graeme W. Austin, Domestic Laws and Foreign Rights: Choice of Law in Transnational Copyright Infringement Litigation, 23 Colum.-VLA J.L. & Arts 1, 18 (1999) (analysing the conflict-of-laws problems that arise when software infringement spans multiple jurisdictions, a problem that has become more acute with cloud-based software distribution).

[49]Council of Europe, Cybercrime Convention, Nov. 23, 2001, E.T.S. No. 185 (Budapest Convention) (the principal multilateral instrument for cross-border cooperation in investigating and prosecuting computer-related offences, including large-scale software piracy).

[50]Stacey Dogan & Mark Lemley, The Trademark Use Requirement in Dilution Cases, 24 Santa Clara High Tech. L.J. 541 (2008); cf. Margaret Jane Radin, Boilerplate: The Fine Print, Vanishing Rights, and the Rule of Law 123-45 (2012) (examining the enforceability of end-user licence agreements and their interaction with copyright exhaustion and first-sale doctrine).